Amendments to the Claims:

The following listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) Method of encoding linguistic frequency data, the method comprising:

mapping each character string occurring in a source text to a numeric identifier identifying the character string,

identifying a plurality of sets of *n* character strings in a source text, *n* being an integer number greater than 1, each set forming an *n*-gram comprising at least a first and a second successive character string,

for each set, obtaining frequency data indicative of the frequency of the respective set in the source text,

creating a memory array *f* for containing the frequency data, *n*-1 pointer memory arrays $p_2 \ldots p_n$ for containing pointers, and *n* offset positional arrays r_1-r_n for containing indexing offsets,

for each character string that is a first character string in at least one of the sets, assigning a memory position in ~~a first~~ memory array *f* to the respective character string and storing at said memory position the frequency data of each set comprising the respective character string as the first character string, ~~and~~

grouping the frequencies relating to *n*-grams that have the same first character string into a block within the array *f*,

for each character string that is a second character string in at least one of the sets, assigning a memory position in ~~a second~~ pointer memory array p_2 to the respective character string and storing at said memory position, for each set comprising the respective character string as the second character string, a pointer pointing to a memory position in the ~~first~~ memory array f assigned to the corresponding first character string of the respective set and having stored the frequency data of the respective set,

grouping the pointers relating to n-grams that have the same second character string together as a block within the pointer memory array p_2,

storing an offset position for each respective first character string in positional array r_1 that indexes to the corresponding block in frequency array f relating to the first character string, and

storing an offset position for each respective i^{th} character string in positional array r_i that indexes to the corresponding block in pointer array p_i relating to the i^{th} character string.

2. (Original) The method of claim 1 wherein each set of character strings further comprises ~~a third~~an i^{th} character string, where $i = 3 \ldots n$, and the method further comprises:

for each character string that is ~~a third~~an i^{th} character string in at least one of the sets, assigning a memory position in ~~a third~~an i^{th} pointer memory array p_i to the respective character string and storing at said memory position, for each set comprising the respective character string as ~~third~~the i^{th} character string, a pointer pointing to a memory position in the ~~second~~ memory array p_{i-1} assigned to the corresponding ~~second~~i^{th}-1 character string of the respective set~~ and having stored a pointer pointing to the frequency data of the respective set~~.

3. (Original) The method of claim 1 wherein each character string is a word of a natural language.

4. (Canceled)

5. (Currently Amended) The method of ~~claim 4~~claim 1 wherein n is equal to 3, the n-grams being trigrams.

6. (Original) The method of claim 1 wherein said frequency data indicative of the frequency of the respective set in the source text includes the number of occurrences of the respective set in the source text.

7. (Original) The method of claim 1 wherein said frequency data indicative of the frequency of the respective set in the source text includes weight numbers of a maximum entropy model.

8. (Original) The method of claim 1, further comprising:

mapping each character string occurring in the source text to a numeric identifier identifying the character string, by operating a finite-state machine.

9. (Canceled)

10. (Original) The method of claim 1, further comprising:

accessing a hash-table for assigning a numeric identifier to each character string occurring in the source text.

11. (Canceled)

12. (Original) The method of claim 1, further comprising:

in the second memory array, sorting the pointers relating to the same second character string, with respect to the memory positions of the first memory array to which the pointers point.

13. (Original) The method of claim 1 wherein the pointers are stored in compressed form.

14. (Currently Amended) Method of accessing encoded linguistic frequency data for retrieving the frequency of a search key in a text encoded according to the method of claim 1, the search key comprising a first and a second search string, the encoded data being stored in ~~a first memory~~ array f storing frequency data and ~~a second~~ memory array p_2 storing

pointers to the ~~first memory~~ array $\underline{f}$, the frequency data being indicative of the frequencies of character sets in ~~a~~the source text, the character sets each including at least two character strings, the method comprising:

identifying a ~~region~~block in the ~~first memory~~ array $\underline{f}$ that is assigned to the first search string,

identifying a ~~region~~block in the ~~second memory~~ array $\underline{p_2}$ that is assigned to the second search string,

identifying a pointer stored in the ~~region~~block of the ~~second memory~~ array $\underline{p_2}$, pointing to a memory position within the ~~region~~block of the ~~first memory~~ array $\underline{f}$, and

reading the frequency data stored at said memory position.

15. (Currently Amended) The method of claim 14 wherein the search key further comprises a third search string and the encoded data is further stored in a ~~third~~ memory array $\underline{p_3}$ storing pointers to the ~~second memory~~ array $\underline{p_2}$, wherein the method further comprises:

identifying a ~~region~~block in the ~~third memory~~ array $\underline{p_3}$ that is assigned to the third search string,

identifying a pointer stored in the ~~region~~block of the ~~third memory~~ array $\underline{p_3}$, pointing to a memory position within the ~~region~~block of the ~~second memory~~ array $\underline{p_2}$, and

tracing the pointer stored in the ~~region~~block of the ~~third memory~~ array $\underline{p_3}$ back until the ~~region~~block of the ~~first memory~~ array $\underline{f}$ is reached.

16. (Original) The method of claim 14 wherein each character string is a word of a natural language.

17. (Original) The method of claim 14 wherein each set of character strings comprising *n* character strings, *n* being an integer number greater than one, each set being an n-gram.

18. (Original) The method of claim 14 wherein *n* is equal to 3, the n-grams being trigrams.

19. (Original) The method of claim 14 wherein said frequency data indicative of the frequency of the respective set in the source text includes the number of occurrences of the respective set in the source text.

20. (Original) The method of claim 14 wherein said frequency data indicative of the frequency of the respective set in the source text includes weight numbers of a maximum entropy model.

21. (Original) The method of claim 14 wherein identifying a pointer includes performing a binary search within the second memory array.

22. (Currently Amended) The method of claim 14 wherein identifying a pointer includes identifying a sub-interval in the ~~region~~block of the ~~second~~ memory array p_2, the sub-interval including at least two pointers pointing to a memory position within the ~~region~~block of the ~~first memory~~ array f.

23. (Original) The method of claim 14, wherein identifying a pointer includes performing a first binary search for a set of pairs of strings where the first string in each pair matches the first search string, performing a second binary search for a set of pairs of strings where the second string in each pair matches the second search string, and calculating an intersection of both sets.

24. (Currently Amended) The method of claim 14, further comprising:

if the character sets in the source text comprise more character strings than the search key and if potentially missing search strings exist, arranging the search strings in the search key such that the potentially missing search strings appear first.

25. (Currently Amended) A system for encoding linguistic frequency data, comprising:

a processing unit for identifying a plurality of sets of character strings in a source text, each set comprising an n-gram including at least a first and a second character string, and, for each set, obtaining frequency data indicative of the frequency of the respective set in the source text, and

an encoder that~~, for each character string that is a first character string in at least one of the sets, assigns a memory position in a first memory array to the respective character string and stores at said memory position the frequency data of each set comprising the respective character string as first character string, and that, for each character string that is a second character string in at least one of the sets, assigns a memory position in a second memory array to the respective character string and stores at said memory position, for each set comprising the respective character string as second character string, a pointer pointing to a memory position in the first memory array assigned to the corresponding first character string of the respective set and having stored the frequency data of the respective set~~:

creates a memory array f for containing the frequency data, n-1 pointer memory arrays $p_2 \ldots p_n$ for containing pointers, and n offset positional arrays r_1-r_n for containing indexing offsets,

for each character string that is a first character string in at least one of the sets, assigns a memory position in memory array f to the respective character string and stores at said memory position the frequency data of each set comprising the respective character string as the first character string,

groups the frequencies relating to n-grams that have the same first character string into a block within the array f, for each character string that is a second character string in at least one of the sets, assigns a memory position in pointer memory array p_2 to the respective character string and stores at said memory position, for each set comprising the respective character string as second character string, a pointer pointing to a memory position in the memory array f assigned to the corresponding first character string of the respective set and having stored the frequency data of the respective set,

groups the pointers relating to n-grams that have the same second character string together as a block within the pointer memory array p_2,

stores an offset position for each respective first character string in positional array r_1 that indexes to the corresponding block in frequency array f relating to the first character string, and

stores an offset position for each respective i^{th} character string in positional array r_i that indexes to the corresponding block in pointer array p_i relating to the i^{th} character string.

26. (Currently Amended) A system for accessing encoded linguistic frequency data encoded by the system of claim 25 for retrieving the frequency of a search key in a text, the search key comprising a first and a second search string, the encoded data being stored in ~~a first memory~~ an array f storing frequency data and ~~a second memory~~ at least one array p_2 storing pointers back to the ~~first memory~~ array f, the frequency data being indicative of the frequencies of character sets in a source text, the character sets each including at least two character strings, the system comprising:

an input device for inputting the search key, and

a search engine for identifying a ~~region~~block in the ~~first memory~~ array f that is assigned to the first search string, identifying a ~~region~~block in the ~~second memory~~ array p_2

that is assigned to the second search string, identifying a pointer stored in the ~~region~~block of the ~~second memory~~ array p_2, the pointer pointing to a memory position within the ~~region~~block of the ~~first memory~~ array *f*, and reading the frequency data stored at said memory position.

27. (New) Method of encoding linguistic frequency data, the method comprising:

mapping each character string occurring in a source text to a numeric identifier identifying the character string,

identifying a plurality of sets of n character strings in a source text, n being an integer number of at least 3, each set forming an n-gram comprising at least a first, a second and a third successive character string,

for each set, obtaining frequency data indicative of the frequency of the respective set in the source text,

creating a memory array f for containing the frequency data, and n-1 pointer memory arrays $p_2 \ldots p_n$ for containing pointers, and n offset positional arrays r_1-r_n for containing indexing offsets,

for each character string that is a first character string in at least one of the sets, assigning a memory position in memory array f to the respective character string and storing at said memory position the frequency data of each set comprising the respective character string as the first character string,

for each character string that is a second character string in at least one of the sets, assigning a memory position in pointer memory array p_2 to the respective character string and storing at said memory position, for each set comprising the respective character string as the second character string, a pointer pointing to a memory position in the memory array f assigned to the corresponding first character string of the respective set and having stored the frequency data of the respective set, and for each character string that is an i^{th} character string in at least one of the sets, for $i = 3..n$, assigning a memory position in an i^{th} pointer memory array p_i to the respective character string and storing at said memory position, for each set comprising the respective character string as the i^{th} character string, a pointer pointing to a memory position in the memory array p_{i-1} assigned to the corresponding i^{th}-1 character string of the respective set,

wherein multiple pointers in the i^{th} pointer memory array p_i point to the same memory position within memory array p_{i-1} and only one chain of memory positions within f, $p_2 \ldots p_n$ uniquely define each n-gram.